PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-360246
(43)Date of publication of application : 14.12.1992

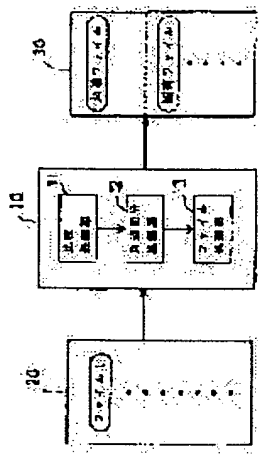
(51)Int.Cl. G06F 12/00

(21)Application number : 03-134694
(22)Date of filing : 06.06.1991

(71)Applicant : TOSHIBA CORP
(72)Inventor : NOZAKI HANAE
ITO SATOSHI

(54) DEVICE FOR COMPRESSING FILE

(57)Abstract:
PURPOSE: To attain efficient file compression by independently storing a common part and a non-common part excluding the common part when common contents are included in plural files.
CONSTITUTION: When a user inputs a file compression processing command and n existing file names required to be compressed, an operating system(OS) mutually compares the contents data of the n files by a comparing processing part 11 and judges whether a completely coincident part of contents exists or not. At the time of judging the existence of the coincident contents part, the part is regarded as a common part, a common part control word indicating the common part is added to the common part by a common part processing part 12 and the common part and its control word are stored in a secondary storage device 30. Then processing for specifying that the compared files have the specific common part on their specific positions is applied to the compared files and the specified files are stored in the device 30. The file processing is repeated for all the n files. Thus efficient file compression can be attained.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平4－360246

(43)公開日　平成４年(1992)12月14日

(51)Int.Cl.⁵	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｇ０６Ｆ　12/00	５１０　Ｂ	8944－5B		

審査請求　未請求　請求項の数１(全 11 頁)

(21)出願番号　特願平3－134694

(22)出願日　平成３年(1991)６月６日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　野崎　華忠
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝総合研究所内

(72)発明者　伊藤　聡
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝総合研究所内

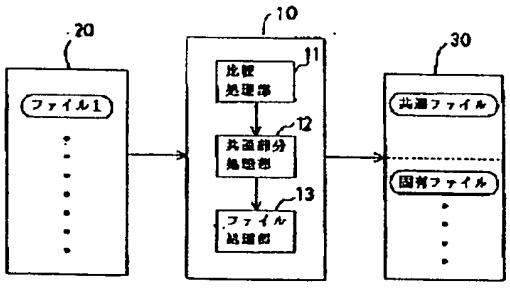
(74)代理人　弁理士　鈴江　武彦

(54)【発明の名称】　ファイル圧縮装置

(57)【要約】

【目的】　複数のファイルに共通の内容を二次記憶装置に重複して記憶する等の不都合を避けることができ、二次記憶装置のより効率的な利用が可能となるファイル圧縮装置を提供すること。

【構成】　共通部分を有する複数のファイルを効率良く圧縮するためのファイル圧縮装置において、磁気ディスク等の二次記憶装置２０に格納された複数のファイルに対しその内容を比較する比較処理部１１と、この比較結果で内容が一致している共通部分を抜き出す共通部分処理部１２と、抜き出した共通部分を共通ファイルとし、さらに共通部分を抜き出した後のファイルを固有ファイルとしてそれぞれ二次記憶装置３０に格納するファイル処理部１３とを備えたことを特徴とする。



1

【特許請求の範囲】

【請求項1】第1の記憶部に格納された複数のファイルに対しその内容を比較する手段と、該手段による比較結果で内容が一致している部分を抜き出す手段と、該手段により抜き出した共通部分を共通ファイルとして第1の記憶部又は第2の記憶部に格納する手段と、前記共通部分を抜き出した後のファイルを固有ファイルとして第1の記憶部又は第2の記憶部に格納する手段とを具備してなることを特徴とするファイル圧縮装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、計算機システムにおいてオペレーティングシステムによって行われるファイル管理の方式に係わり、特に共通部分を有する複数のファイルを効率良く圧縮するためのファイル圧縮装置に関する。

【0002】

【従来の技術】計算機システムにおいて、記憶領域である二次記憶装置を効率的に使用してデータの蓄積経費の削減を行い、かつ転送経費の経済化をはかるため、データ圧縮技術が提案され実用化されている。データ圧縮とは、データ変換を行うことによってデータ内の冗長度を抑圧し、データをより短いデータ長で簡潔に表現することである。

【0003】現在、広く使用されているデータ圧縮法として、Ｈｕｆｆｍａｎの最適符号化法がある。これは、データ中のパターンの出現頻度を統計的に調べ、出現頻度の高いパターンほど短い符号を割り当てるもので、パターンの数を多くすることにより平均符号長を短くすることが可能となる。また、Ｚｉｖ－Ｌｅｍｐｅｌのデータ圧縮法では、データの統計的性質や定常性を仮定する必要がなく、任意の記号列に直接適用できる。このデータ圧縮法によると、長い記号列を効率良く圧縮することが可能なため、計算機システムで作られる各種ファイルの圧縮に適している。

【0004】しかしながら、この種のデータ圧縮技術にあっては次のような問題があった。即ち、複数のファイルが等しい内容のデータを有している場合、記憶領域（二次記憶装置）には同じデータがいくつも保存されることになる。この二次記憶装置における無駄な領域は、内容が共通である部分のサイズが大きいほど、またそれを有するファイル数が多いほど大きくなり、二次記憶装置の効率的な利用の妨げとなっている。

【0005】一方、上記のようにデータ圧縮処理は、データファイルに対して個別に実行され、個々のデータファイルがよりサイズの小さいデータファイルに変換される。そのため、複数のファイルに共通の内容が内在していることによる二次記憶装置利用上の不経済性を、従来のデータ圧縮処理では排除することができない。

【0006】また、この問題点を解決するには、複数のファイルの内容を考慮した統合的な処理が必要であるが、これまでこのような処理方法は全く実施されておらず、未だ実用化されていない。

2

【0007】

【発明が解決しようとする課題】このように従来のデータ圧縮処理は、個々のデータファイルに関してはそれぞれ圧縮効果があるものの、複数のファイルに共通の内容が内在していることによる二次記憶装置利用上の不経済性を排除することはできなかった。

【0008】本発明は、上記問題点を解決するためになされたもので、その目的とするところは、複数のファイルに共通の内容を二次記憶装置に重複して記憶する等の不都合を避けることができ、二次記憶装置のより効率的な利用が可能となるファイル圧縮装置を提供することにある。

【0009】

【課題を解決するための手段】本発明の骨子は、複数のファイルに共通の内容がある場合に、共通部分とこの共通部分を除いた非共通部分とを独立して格納することにある。

【0010】即ち本発明は、複数のファイルを効率良く圧縮するためのファイル圧縮装置において、第1の記憶部に格納された複数のファイルに対しその内容を比較する手段と、この比較結果で内容が一致している部分を抜き出す手段と、抜き出した共通部分を共通ファイルとして第1の記憶部又は第2の記憶部に格納する手段と、共通部分を抜き出した後のファイルを固有ファイルとして第1の記憶部又は第2の記憶部に格納する手段とを設けるようにしたものである。

【0011】本発明においては、複数のファイルの内容を比較した結果、一つのファイル内で判定される共通部分が複数個あってもよい。また、比較はユーザーが指定した全てのファイル間で行うだけに限らず、指定したファイルの内の任意の複数ファイルに対して比較処理を行い、そのファイル間の共通部分を判定するようにしてもよい。

【0012】

【作用】本発明によれば、以下のような状況において非常に効率の良いファイル圧縮を行うことができる。

【0013】例えば、系の時間発展を調べるため計算機によるシミュレーションが行われているものとする。このようなシミュレーションでは、時間の離散化を行い、その離散化された各タイムステップ毎に系の状態を計算して時間発展を追う。計算された各タイムステップ毎の現象の変化を把握するには、シミュレーション結果のグラフィック化が非常に有効な支援手段であるため、通常、あるタイムステップ間隔でグラフィック用のデータをアウトプットしてデータファイルを作成する。そして、そのデータファイルを元に、一タイムステップ毎の静止画をつなげて動画を作る。

3

【００１４】一般にグラフィック用としてアウトプットされるデータ量は、一つのタイムステップ分でも大きく、データファイル全体のサイズは膨大なものとなる。ところで、シミュレーションの特徴の一つとして、実行者が希望する初期値，条件のもとでの系の状態の追跡が可能であるという恣意性をあげることができる。そのため、同じ初期値において条件を変えて何回も計算を行い、条件の違いによる時間発展の違いを調べるというシミュレーションの実行例が考えられる。

【００１５】このように、同じ初期値で条件を変えたシミュレーションを繰り返し行い、その結果をグラフィック化するためにデータファイルを作成した場合、データファイルにアウトプットされる初期データ（一タイムステップ分のデータであり，動画における初期画面となる）は全てのデータファイルで共通のものとなる。しかし、前述したようにグラフィック用のデータのサイズは一つのタイムステップでも非常に大きなものであるため、いくつものデータファイルが同じ初期データを共有していることは、二次記憶装置中に無駄な領域を作り出すことになり、記憶装置の効率的な利用の妨げとなっている。

【００１６】このような場合、本発明のファイル圧縮処理を行えば、初期データはただ一つ保存されることになるため、二次記憶装置の効率的な使用が可能となり、かつユーザーは圧縮処理の行われたデータファイルを、それぞれが初期データを保持しているものとして扱うことができる。言うまでも無く、初期データがファイル間で共通である場合、全てのデータファイルに初期データをアウトプットする必要はないのであるが、系の時間発展の様子を視覚的に的確に捕らえ、そこに現れている物理的，化学的現象をより良く理解するためには、グラフィック用のデータファイルに初期データが含まれていることが非常に望ましい。

【００１７】また、初期データとそれ以降のタイムステップのデータとを分割して、別のファイルにアウトプットすることも可能であるが、このようにデータファイルを分けた場合、その後のグラフィック化の処理やデータファイルの管理がかなり煩雑なものとなる。そのため，ファイル圧縮処理は二次記憶装置の有効利用のためのみでなく，ユーザーが計算機システムを能率的に使用する上でも効果が発揮されるといえる。

【００１８】その他，ファイル圧縮処理が有効である例として、各時刻ｔでのｘの値を計算してｔとｘの値を出力するプログラムの実行が考えられる。このプログラムを、時間刻みは同じで条件を変えて何回も実行した場合、データファイルに出力される時刻ｔに関する数列は、作成される全てのデータファイルで等しくなる。従って、ｔの値という共通の内容を含むこれらのデータファイルに対しても、このファイル圧縮処理は効果を持つ。また、僅かな変更で実現できるような新しい機能や

4

異なる機能を実行可能なプログラムに新たに持たせたい場合、そのソースファイルをコピーし、修正して使用する。このような場合にも共通の内容を含むファイルがいくつか作られることになるため、ファイル圧縮処理を行うことで二次記憶装置の有効利用に貢献することができる。

【００１９】また、全てのファイル間で行うだけに限らず、指定したファイルの内の任意の複数ファイルに対して比較処理を行うことにより、圧縮効率が最大となるように処理することができる。例えば、３個のファイルが存在しているときに、第１のファイルに対し第２のファイルは共通部分が多く、第３のファイルは共通部分が極めて少ないとする。この場合、全てのファイルを比較処理すると，共通部分は第１と第３のファイルの共通部分のみとなり、圧縮効率は極めて低くなる。これに対し、第１と第２のファイルを指定して比較処理を行えば、共通部分のデータ量が多くなり、圧縮効率の向上をはかることが可能となる。

【００２０】

【実施例】以下、本発明の実施例を図面を参照して説明する。

【００２１】図１は、本発明の一実施例に係わるファイル圧縮装置の概略構成を示すブロック図である。図中１０は本発明に係わるファイル圧縮処理部であり、複数のファイルを比較する比較処理部１１，比較したファイルの共通部分を取り出す共通部分処理部１２，及び各ファイルを共通部分と非共通部分に分けるファイル処理部１３等から構成されている。２０は磁気ディスク等の二次記憶装置（第１の記憶部），３０も同様に磁気ディスク等の二次記憶装置（第２の記憶部）である。なお、これらの二次記憶装置２０，３０は必ずしも独立したものではなく、共通のものであってもよい。上記装置によるファイル圧縮の動作を、図２～図１３に示すフローチャート及びファイル構造を参照して説明する。

【００２２】まず、基本的なファイル圧縮処理の手順を、図２に示すフローチャートに従って説明する。ユーザーがファイル圧縮処理のコマンドと圧縮を希望する既存のｎ個のファイル名を入力する（ステップＳ１）と，オペレーティングシステムは指定されたｎ個のファイル間で内容データの比較を行い（ステップＳ２），内容が完全に一致している部分があるか否かを判断する（ステップＳ３）。内容が一致している部分がなかった場合，ファイル圧縮処理を終了する。

【００２３】一方、内容が一致している部分があると判定された場合には、その部分を共通部分とみなし、それに共通部分であることを示す共通部分制御語を付加して二次記憶装置に保存する（この処理を共通部分処理と呼ぶ）（ステップＳ４）。さらに、比較が行われた後のファイルに対して、そのファイルが特定の位置に特定の共通部分を所有していることを指定するための処理を行

5

い、二次記憶装置に保存する（この処理を比較後ファイル処理と呼ぶ）（ステップＳ５）。この比較後ファイル処理をｎ個のファイル全てに対して行うと、ファイル圧縮処理が終了される。ファイル圧縮処理によって新たに作成されたファイルを共通部分も含めて圧縮済ファイルと呼ぶ。

【００２４】ここで、ファイル間の内容データを比較して共通部分の判定を行う方法を簡単に説明する。例として、ファイル１とファイル２の共通部分を判定する場合を考える。まず、ファイル１を適当なデータ長（例えばこの場合１行とする）に分割したとみなし、ファイル１の全行とファイル２の全行を先頭行から順に１行ずつ比較する。あるペアを比較した結果、その二つの行が同一でなければ次のペアの比較を行うが、同一であった場合には以下の処理を行う。例えば、ファイル１の i_1 行目とファイル２の i_2 行目が同一であった場合は、次にファイル１の（i_1+1）行目とファイル２の（i_2+1）行目の比較を行う。この二つの行も同じであれば、（i_1+2）行目と（i_2+2）行目を比較する。

【００２５】このように、二つの行が同一である限りファイル１とファイル２のそれぞれ次の行の比較を行う。そして、仮にファイル１の（i_1+k）行目とファイル２の（i_2+k）行目が同一でないと判明した場合、ファイル１の i_1 行目から（i_1+k-1）行目までの範囲とファイル２の i_2 行目から（i_2+k-1）行目までの範囲が共通部分であると判定される。もし、３個のファイルの共通部分を判定する場合は、ファイル１とファイル２の共通部分とファイル３を上記と同様の方法で比較する。

【００２６】次に、共通部分処理と比較後ファイル処理について、より具体的に説明する。まず、図２のステップＳ４の共通部分処理では、上述したように共通部分に共通部分制御語を付加し、図３のようなファイル（これを共通ファイルと呼ぶ）として二次記憶装置に保存する。ここで、共通部分制御語にはその共通部分を共有している圧縮済ファイル数（これを共有ファイル数と呼ぶ）を記録する。この共通部分制御語は、後に説明するように圧縮済ファイルの更新，削除等を行うために設ける。

【００２７】図２のステップＳ５の比較後ファイル処理として、インデックス逐次型ファイル方式を用いることができる。まず、従来用いられているインデックス逐次型ファイルについて説明する。インデックス逐次型ファイルとは、一つのインデックスといくつかのレコード（ファイルを分割したもの）で構成された構造を持ち，インデックスにはレコードを指定するための値（レコードが保存されている二次記憶装置中の先頭位置と末尾位置）が順番に記憶されている。オペレーティングシステムはインデックスからレコードに関する情報を読取り，順番にレコードにアクセスすることによって、インデックス逐次型ファイルの処理を行うことができる。これによると、レコードの更新，挿入，削除も可能である。

6

【００２８】本発明の比較後ファイル処理では、このインデックス逐次型ファイル方式を応用して、オペレーティングシステムは図４に示すフローチャートに従った処理を行う。初めに、比較が行われた後のファイルを共通部分とそれ以外の部分（これを固有部分と呼ぶ）に分割し、それぞれをレコードとみなす（ステップＳ６）。つまり、レコードは共通部分或いは固有部分のどちらかであるが、共通部分である場合，共通部分制御語を含む共通ファイルをレコードとみなす。次に、新たに設けたインデックスにそれぞれのレコードの先頭位置と末尾位置を示す値と、そのレコードが共通部分であるか固有部分であるかを区別する値を記録する（ステップＳ７）。最後に、インデックスと固有部分であるレコードを二次記憶装置に保存して（ステップＳ８）、比較後ファイル処理が終了される。

【００２９】従って、このインデックス逐次型ファイル方式でファイル圧縮処理を行った場合、圧縮済ファイルは図５に示すような構造を持つ。このようにして作成された圧縮済ファイルに対して、通常のインデックス逐次型ファイルに対する処理（更新，削除等）と同様の処理を行うことが可能である。但し、レコードが共通ファイルである場合、レコードの先頭に共通部分制御語が付加されているため、共通部分制御語を読み飛ばし、共通部分のみを処理する。

【００３０】また、比較後ファイル処理をマーキング方式と呼ぶ処理方法を用いて実行することも可能である。この方式では、オペレーティングシステムは図６に示すフローチャートに従った処理を行う。まず、比較処理が行われたファイルから共通部分を取り除き、その位置に共通部分が存在することを示すマーキングを行う（ステップＳ９）。次に、ファイルにヘッダーを付加し、それにファイルから取り除いた共通部分に関する情報を記録する（ステップＳ１０）。ステップＳ９とステップＳ１０によって処理されたファイルを固有ファイルと呼び，その構造を図７に示す。最後に、固有ファイルを二次記憶装置に保存して（ステップＳ１１）、比較後ファイル処理の終了となる。

【００３１】具体的なマーキングの方法として、共通部分が抜き出された位置に特殊文字（これを共通部分指定文字と呼ぶ）を記録し、その後ろに抜き出した共通部分（共通ファイル）を指定する値を記録する。この値は、例えば共通ファイルが保存されている二次記憶装置中の先頭位置と末尾位置を示す値である。ここで、共通部分指定文字とそれに続く値を共通部分指定語と呼ぶ。また、ヘッダーにも共通ファイルが保存されている先頭位置と末尾位置を示す値を記録する。従って、マーキング方式によるファイル圧縮処理で作成される圧縮済ファイルは一つの固有ファイルとそれが指定する共通ファイル

7

から構成される。例えば、オペレーティングシステムが[illegible]によって作成された圧縮済ファイルを更新する場合、次のような処理を行うことで圧縮済ファイルを通常のファイルと同様に扱うことができる。

【0032】即ち、圧縮済ファイルの更新のためには、固有ファイルが二次記憶装置から作業領域である主記憶或いは二次仮想アドレス空間へ転送される。転送の際、各ビット毎にそれが共通部分指定文字であるか否かの判定を行い、共通部分指定文字であると判定された場合は、共通部分指定語をそれが指定する共通部分と置き換える。つまり、共通部分指定語を取り除き、その位置に共通部分の読み込みを行う。この処理によって、作業領域に転送される固有ファイルは、ファイル圧縮処理を行う前と同一のファイルに変換される。但し、固有ファイルのヘッダーの転送は行わない。

【0033】以上記述してきた方法によると、本発明のファイル圧縮処理を実現することが可能となる。次に、上述した方法によって作成された圧縮済ファイルを削除あるいは更新するための処理と、圧縮済ファイルに対しさらにファイル圧縮処理を行う場合の具体的な手順について説明する。

【0034】インデツクス逐次型ファイル方式によって作成された圧縮済ファイルを削除する場合、オペレーティングシステムは図8に示すフローチャートに従った処理を行う。ユーザーが削除のコマンドと削除したい圧縮済ファイルの名前を入力すると（ステップS12）、オペレーティングシステムは指定された圧縮済ファイルのインデックスにアクセスし（ステップS13）、レコードが共通部分であるか固有部分であるかの判定を行う（ステップS14）。レコードが固有部分である場合、オペレーティングシステムはそのレコード（固有部分）の削除を行う（ステップS15）。

【0035】一方、レコードが共通部分である場合、オペレーティングシステムはその共通ファイルの共通部分制御語にアクセスし、共有ファイル数の書き替えを行う（ステップS16）。この場合は共有ファイル数を1つ減らす。次に、共有ファイル数が0であるかないかの判定を行い、共有ファイル数が0であればその共通部分を削除する（この処理を零判定削除と呼ぶ）（ステップS17）。ユーザーが指定した圧縮済ファイルが所有する全てのレコードに対して、ステップS13からステップS17までの処理が終わると、最後にインデックスが削除され（ステップS18）、圧縮済ファイルの削除が完了する。

【0036】マーキング方式による圧縮済ファイルを削除する場合は、図9のフローチャートに従った処理が行われる。まず、ユーザーが削除のコマンドと削除したい圧縮済ファイルの名前を入力すると（ステップS19）、オペレーティングシステムは指定された圧縮済ファイルの固有ファイルのヘッダーにアクセスし、共通ファイルに関する情報を読み取る（ステップS20）。次に、共通ファイルの共有ファイル数を1つ減らして（ステップS21）、零判定削除を行う（ステップS22）。最後に、固有ファイルを削除して（ステップS23）、圧縮済ファイルの削除の終了となる。

8

【0037】インデックス逐次型ファイル方式による圧縮済ファイルに対する更新の手続きを、図10に示すフローチャートに従って説明する。ユーザーは圧縮済ファイルの編集のコマンドと圧縮済ファイル名を入力して（ステップS24）、圧縮済ファイルの編集を行う。（ステップS25）。編集が終った圧縮済ファイルを保存する際、オペレーティングシステムはレコード毎にその内部で変更が行われたかをチェックする（ステップS26）。レコード内で変更がなされなかった場合、そのレコードに対する処理は何も行わない。しかし、レコード内で変更が行われた場合、レコードが共通部分であるか、固有部分であるかを判定し（ステップS27）、レコードが固有部分であればそれを更新する（ステップS28）。

【0038】一方、変更されたレコードが共通部分である場合、そのレコードを固有部分として更新する（ステップS29）。つまり、共通部分制御語を取り除いた状態で新たに保存する。そして、インデクッスの該当するレコードの情報を、新たに保存したレコードを指定する値に書き替える（ステップS30）。さらに、変更される前の共通ファイルに対しては、共通部分制御語内の共有ファイル数を1つ減らして（ステップS31）、零判定削除を行う（ステップS32）。ステップS26からステップS32までの処理を圧縮済ファイルが所有する全てのレコードに対して行って、圧縮済ファイルの更新が完了となる。

【0039】マーキング方式による圧縮済ファイルの更新を行う場合、変更したファイルは通常のファイルとして保存する。そして、変更前の圧縮済ファイルが所有していた共通ファイルの共有ファイル数を1つ減らし、零判定削除を行う。従って、マーキング方式による圧縮済ファイルに一旦更新が行われると、それはもはや圧縮済ファイルではなく、通常のファイルに戻る。

【0040】圧縮済ファイルに対するファイル圧縮処理は、圧縮済ファイルを通常のファイルに戻した後、改めて行う。圧縮済ファイルを通常のファイルに戻す処理は次のような手順で行う。インデックス逐行型ファイル方式で処理された圧縮済ファイルの場合、インデックスに従ってレコードを一つのファイルにつなげ直す。その際レコードが共通ファイルであれば、共通部分制御語を除いた共通部分のみをつなげる。マーキング方式による圧縮済ファイルの場合、共通部分指定語をそれが指定する共通部分と置き換える。そして、圧縮済ファイルが所有していた共通ファイルの共有ファイル数を1つ減らし、零判定削除を行う。最後に、インデックス逐次型ファイ

9

ル方式の場合はインデックスを、マーキング方式の場合はヘッダー部分を消去する。以上の手続きにより圧縮済ファイルは通常のファイルに戻るため、その通常のファイルに対して改めてファイル圧縮処理を行う。

【００４１】これまで、処理される共通部分が１個であるという前提の下でファイル圧縮処理の説明を行ってきたが、共通部分が複数個ある場合でもファイル圧縮処理を実現するにあたり、何等支障を来すものではない。そのため、判定される共通部分が複数個ある場合の処理方法について説明する。

【００４２】ｎ個のファイル中に、例えばｋ個の共通部分が含まれている場合のファイル圧縮処理は、基本的には図２のフローチャートと同様の手続きでよいが、一部異なる箇所がある。図２のステップＳ２においてｎ個のファイルが比較された結果、共通部分がｋ個判定された場合、ステップＳ４がｋ回繰り返され、判定されたｋ個の共通部分それぞれについて共通部分処理が行われる。次に、ｎ個のファイルに対して比較後ファイル処理（ステップＳ５）が行われるが、インデックス逐次型ファイル方式の場合、レコードのうちｋ個が共通部分になっており、マーキング方式の場合、固有ファイルにはｋ個の共通部分指定語が記録される。またインデックスとヘッダーには、ｋ個の共通部分全ての情報が記録される。この比較後ファイル処理がｎ回繰り返され、ｎ個のファイル全てに対する処理が終わると、ファイル圧縮処理が終了する。

【００４３】比較処理はユーザーが指定したｎ個ファイル全てに対して行うだけに限らず、その内の任意のファイル間で行われるものであってもよい。そのため、ファイル圧縮処理を行う際、比較するファイル数ｉを変化させて、段階的に比較処理と共通部分処理を実行する。つまり、段階ｉにおいてはｉ個（ｉ≦ｎ）のファイル間で比較を行い、そのｉ個のファイルに対する共通部分ｉを判定して、共通部分処理を行う。ｉの値はｎからｍ（ｎ≧ｍ≧２）まで減少方向へ変化させる。

【００４４】図１１のフローチャートを参照して、具体的な処理の手順を説明する。ユーザーが、ファイル圧縮処理のコマンドとｎ個のファイル名を入力する（ステップＳ３３）と、まずオペレーティングシステムはｎ個のファイルに対して比較処理を行い（ステップＳ３４、ｉ＝ｎ）、全ファイルの共通部分（共通部分ｎ）を判定し（ステップＳ３５）、共通部分ｎがあれば共通部分処理を行う（ステップＳ３６）。次に、任意のｎ－１個のファイル間において、共通部分ｎを除いた範囲で内容の比較を行い（ステップＳ３４、Ｉ＝ｎ－１）、共通部分ｎ－１があれば、共通部分処理を実行する（ステップＳ３６）。但し、ステップＳ３４からステップＳ３６までの処理は、ｎＣｌ（ｎ個からｌ個選び出す組み合わせの数）回繰り返す。つまり、ｉがｎ－１の時、繰り返しの回数はｎC_{n-1}＝ｎであり、全ての組み合わせのｎ－１

10

個のファイル間で、比較処理と共通部分処理を行う。

【００４５】このようにして、最終的にはｍ個（ｎ≧ｍ≧２）のファイル間での比較処理と共通部分処理まで段階別に処理を行う。最後に、入力したｎ個のファイルに対して、そのファイル中で判定された全ての共通部分に関する比較後ファイル処理を行い（ステップＳ３７）、ファイル圧縮処理が終了される。以下、ファイル圧縮処理に関する補助的な事項について記述する。

【００４６】(1) ファイル間の比較の結果、判定される共通部分のサイズが非常に小さい場合は、ファイル圧縮処理の効果が十分得られないため、共通部分サイズの下限値の指定が必要である。従ってこの場合、例えば下限値を共通部分制御語のサイズと定める。

【００４７】(2) 共通部分処理において、共通ファイルの共通部分制御語に記録する情報は共有ファイル数であるが、加えて、その共通ファイルを共有している圧縮済ファイルの名前とその記憶位置を記録してもよい。

【００４８】(3) インデックス逐次型ファイル方式による比較後ファイル処理において、レコードのサイズが非常に大きい場合、それをさらにいくつかのレコードに分割してもよい。

【００４９】(4) マーキング方式による比較後ファイル処理において、固有ファイルの共通部分指定語中で共通部分指定文字に続いて記録される値は、共通ファイルを指定できる値であればよいため、例えば共通ファイルが保存されている先頭位置と末尾位置を示す値を記録するか、若しくは共通ファイルの名前を記録してもよい。但し、共通部分指定語に共通ファイルの名前を記録した場合、ヘッダーには共通ファイルの先頭位置と末尾位置を示す値に加えて、共通ファイルの名前も記録しなければならない。

【００５０】(5) オペレーティングシステムはファイル圧縮処理を終了する前に、ユーザーに対して処理状況を知らせてもよい。例えば、ユーザーがオンラインで操作を行っている場合、端末の画面に処理状況をメッセージとして表示する。即ち、共通部分が判定された場合、判定された共通部分の名前或いは番号とその共通部分を共有する圧縮済ファイル名のリストを、共通部分が判定されなかった場合、共通部分がなかったという内容のメッセージを表示してもよい。

【００５１】(6) 比較するファイル数を減少させながら、共通部分の判定と共通部分処理を段階的に行う場合のファイル圧縮処理（図１１）において、最終段階で比較されるファイルの個数ｍは暗黙に指定がなされていてもよいし、ユーザーが指定し直すことができてもよい。

【００５２】(7) ｎ個のファイル間の共通部分ｎを判定する方法として、例えば次のような手順がある。ｎ個のファイルに番号付けを行い、まず１番目と２番目のファイルを比較して、共通部分を抽出する。次に、抽出した共通部分と３番目のファイルを比較して、同じく共通部

11

分を抽出する。このようにファイルを一つずつ順番に比較していき、最後のｎ－１個のファイル間での共通部分とｎ番目のファイルを比較して、共通部分ｎを判定する。よって共通部分ｎが判定された場合、１番目からｉ番目（$2 \leq i \leq n-1$）までのファイル間では、既に比較が行われ共通部分が抽出されている。ここで、共通部分ｎを判定する過程で抽出されるこれらの共通部分を仮共通部分と呼ぶ。ところで、共通部分の判定と共通部分処理を段階的に行う場合のファイル圧縮処理（図１１）では比較するファイル数を減少方向に変化させるため、共通部分ｉを判定する段階では、共通部分ｎから共通部分ｉ＋１までは判定済みとなっている。従って、上記の方法で共通部分ｎを判定している場合には、幾つかの仮共通部分が既に抽出されているため、ファイル圧縮処理のステップ数を減らす目的で、抽出済みの仮共通部分を共通部分ｉの判定のために再利用してもよい。

【００５３】(8) １個のファイル内に共通部分が幾つか存在する場合、このファイルに対し単独でファイル圧縮処理を行ってもよい。これを実現するためには、ユーザーがファイル圧縮処理のコマンドとファイル名を一つ入力し、一つのファイル内だけで内容の比較を行って共通部分１を判定してもよいし、或いは図１１のファイル圧縮処理において、最終的に比較するファイル数ｍを１としてもよい。

【００５４】(9) 本発明におけるファイル圧縮方式は、従来行われているデータ圧縮技術（Huffman の最適符号化法、Ziv -Lempel のデータ圧縮法など）とは全く異質の処理方法であるため、ファイル圧縮処理を行った後、圧縮済ファイルに対してさらに従来のデータ圧縮を行うことに何等問題はない。さらに、ファイル圧縮処理における変形例をいくつか説明する。

【００５５】圧縮済ファイルに対する更新処理の特殊な場合として、共通部分の更新が考えられる。これは共通部分そのものを書き替えてしまう処理であり、この処理を行うことにより、その共通部分を含む圧縮済ファイルは共通部分の内容が一斉に変更されることになる。共通部分を更新するためには、共通部分制御語の変更は行わず、そのまま更新を行えばよい。このように、同じ内容を含んでいる複数のファイルに対して、その共通部分の一括変更を希望する場合、ファイル圧縮処理と共通部分の更新を総合した処理（これをファイル一括変更と呼ぶ）を行うと非常に便利である。

【００５６】ファイルの一括変更は、図１２に示すフローチャートに従って行われる。初めに、ユーザーはファイル一括変更のコマンドと一括変更を希望するｎ個のファイル名を入力する（ステップＳ３８）。オペレーティングシステムはそれらのファイルに対しファイル圧縮処理を行い（ステップＳ３９）、共通部分の名前或いは番号とそれを共有している圧縮済ファイル名のリストを表示する（ステップＳ４０）。次に、ユーザーは更新した

12

い共通部分の編集を行い（ステップＳ４１）、オペレーティングシステムはその修正された共通部分を更新する（ステップＳ４２）。この共通部分の修正と更新は繰り返し行うことが可能である。更新したい共通部分がなくなれば処理を終了する。

【００５７】また、ファイル圧縮処理をコピー処理として用いることも可能である。この処理の手順を図１３に示すフローチャートを参照して説明する。まず、ユーザーはコピー（ファイル圧縮処理による）のコマンドとコピー元とコピー先のファイル名を入力する（ステップＳ４３）。オペレーティングシステムはコピーするファイル全体を共通部分とみなし（ステップＳ４４）、共通部分処理（ステップＳ４５）において、共有ファイル数を２と記録する。さらに、比較後ファイル処理（ステップＳ４６）を２回行って、コピー元とコピー先の圧縮ファイルを作成する。

【００５８】インデックス逐次型ファイル方式による比較後ファイル処理の場合、図４のステップＳ６の処理は必要はなく、インデックスに記録される情報は一つのレコード（共通ファイル）の情報のみである（ステップＳ７）。そして、ステップＳ８ではインデックスのみが保存される。また、マーキング方式による比較後ファイル処理の場合、図６に示したフローチャートに従って処理を行う。但し、この比較後ファイル処理で作成される固有ファイルはヘッダーと共通部分指定語のみから構成される。以上のような処理を行うことによって、二次記憶装置には共通部分がただ一つ保存され、それをコピー元とコピー先の圧縮済ファイルが共有することになる。

【００５９】なお、本発明は上述した実施例に限定されるものではない。実施例ではファイルを格納する記憶部として磁気ディスクを用いたが、この代わりには磁気テープや光ディスク等の二次記憶装置（外部記憶装置）を用いることができる。また、図１に示すファイル圧縮処理部の構成は、ハードウェアによって実現してもよいし、ソフトウェアによって実現してもよい。その他、本発明の要旨を逸脱しない範囲で、種々変形して実施することができる。

【００６０】

【発明の効果】以上詳述したように本発明によれば、内容が共通した部分を含むファイルが複数存在する場合に、各ファイルを共通部分とこの共通部分を除いた非共通部分とに分け、これらを二次記憶装置等の記憶部に独立して格納している。従って、複数のファイルに共通の内容を二次記憶装置に重複して記憶する等の不都合を避けることができ、二次記憶装置のより効率的な利用が可能となるファイル圧縮装置を実現することが可能となる。

【図面の簡単な説明】

【図１】本発明の一実施例に係わるファイル圧縮装置の概略構成を示すブロック図、

13

【図2】ファイル圧縮処理の基本的な手順を示すフローチャート、

【図3】共通ファイルの構造を示す模式図、

【図4】インデックス逐次型ファイル方式による比較後ファイル処理を示すフローチャート、

【図5】インデックス逐次型ファイル方式によるファイル圧縮処理で作成される圧縮ファイルの構造を示す模式図、

【図6】マーキング方式による比較後ファイル処理を示すフローチャート、

【図7】マーキング方式によるファイル圧縮処理で作成される圧縮ファイルが持つ固有ファイルの構造を示す模式図、

【図8】インデックス逐次型ファイル方式で処理された圧縮ファイルを削除する手順を示すフローチャート、

【図9】マーキング方式で処理された圧縮ファイルを削除する手順を示すフローチャート、

14

【図10】インデックス逐次型ファイル方式で処理された圧縮ファイルを更新する手順を示すフローチャート、

【図11】比較するファイル数を変化させ、段階的に比較処理と共通部分処理を行う場合のファイル圧縮処理の手順を示すフローチャート、

【図12】ファイル一括変更の手順を示すフローチャート、

【図13】ファイル圧縮処理をコピー処理として行う場合の手順を示すフローチャート。

10

【符号の説明】

10…ファイル圧縮処理部、

11…比較処理部、

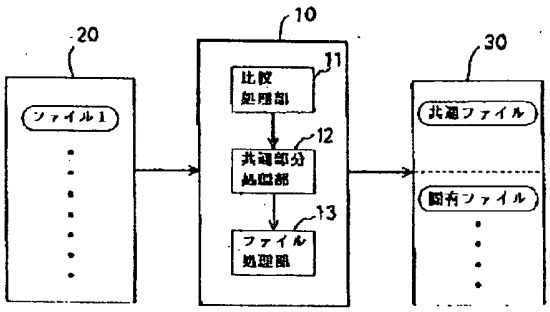
12…共通部分処理部、

13…ファイル処理部、

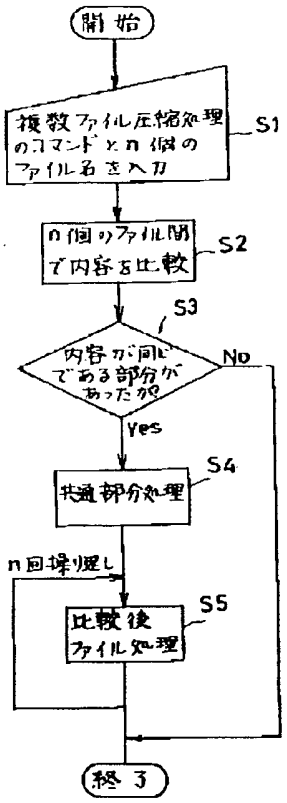
20…二次記憶装置（第1の記憶部）、

30…二次記憶装置（第2の記憶部）。

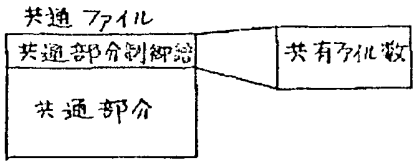
【図1】



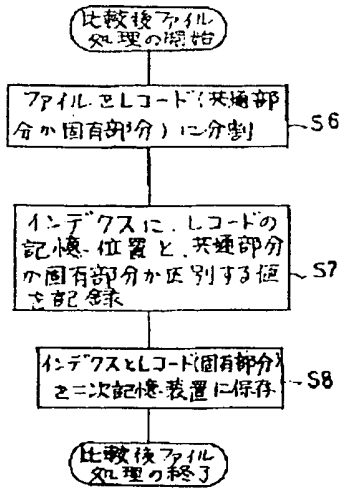
【図2】



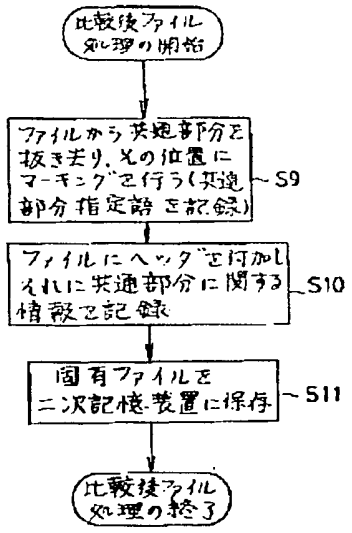
【図3】


共通ファイル
共通部分制御語
共有ファイル数
共通部分

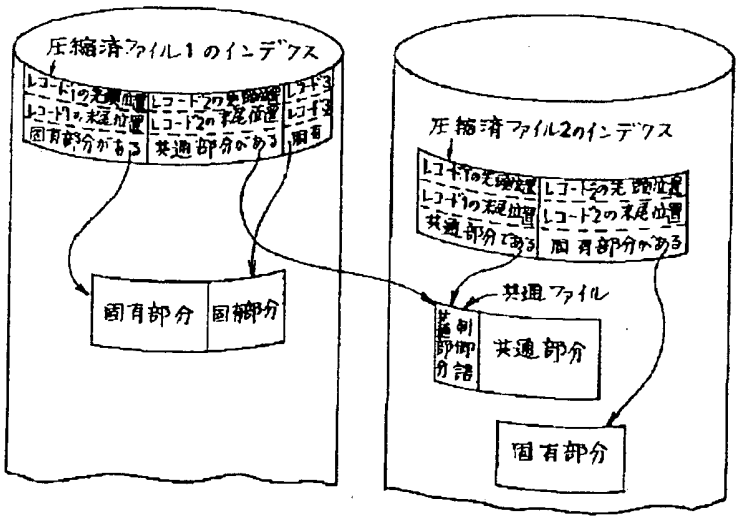
【図4】


比較後ファイル処理の開始
ファイルをレコード(共通部分か固有部分)に分割
S6
インデクスに、レコードの記憶位置と、共通部分か固有部分か区別する値を記録
S7
インデクスとレコード(固有部分)を二次記憶装置に保存
S8
比較後ファイル処理の終了

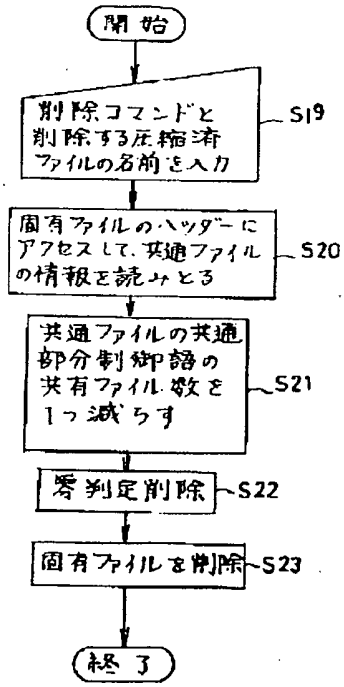
【図6】


比較後ファイル処理の開始
ファイルから共通部分を抜き去り、その位置にマーキングを行う(共通部分指定語を記録)
S9
ファイルにヘッダを付加しそれに共通部分に関する情報を記録
S10
固有ファイルを二次記憶装置に保存
S11
比較後ファイル処理の終了

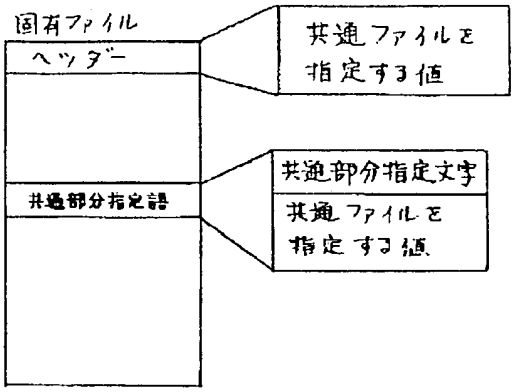
【図5】


圧縮済ファイル1のインデクス
レコード1の先頭位置
レコード1の末尾位置
固有部分がある
レコード2の先頭位置
共通部分がある
固有部分
固有部分
圧縮済ファイル2のインデクス
レコード1の先頭位置
レコード1の末尾位置
共通部分である
レコード2の先頭位置
固有部分がある
共通ファイル
共通部分制御語
共通部分
固有部分

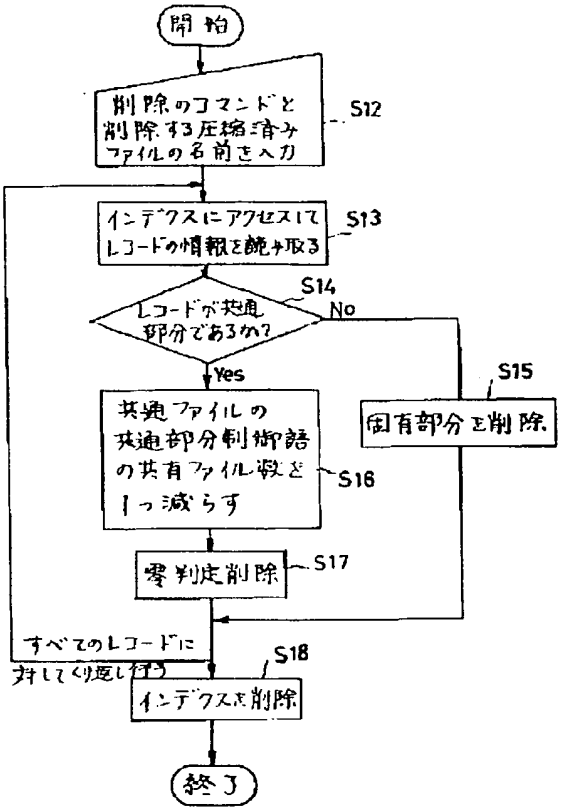
【図9】


開始
削除コマンドと削除する圧縮済ファイルの名前を入力
S19
固有ファイルのヘッダーにアクセスして、共通ファイルの情報を読みとる
S20
共通ファイルの共通部分制御語の共有ファイル数を1つ減らす
S21
零判定削除
S22
固有ファイルを削除
S23
終了

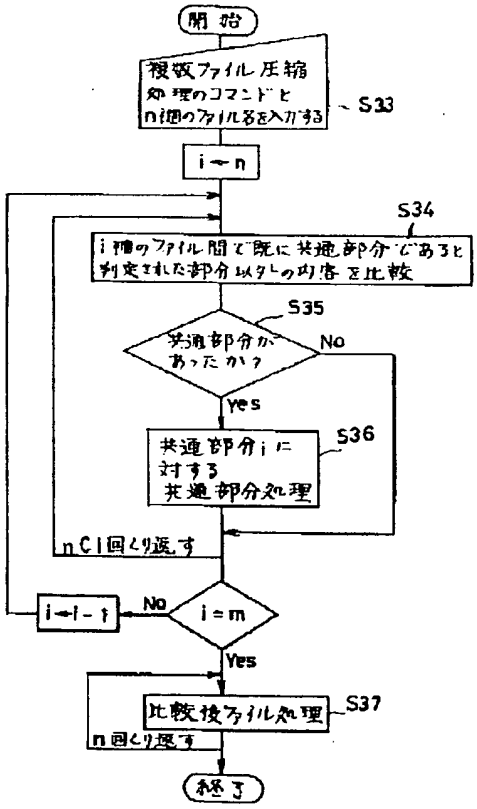
【図7】


固有ファイル
ヘッダー
共通ファイルを指定する値
共通部分指定語
共通部分指定文字
共通ファイルを指定する値

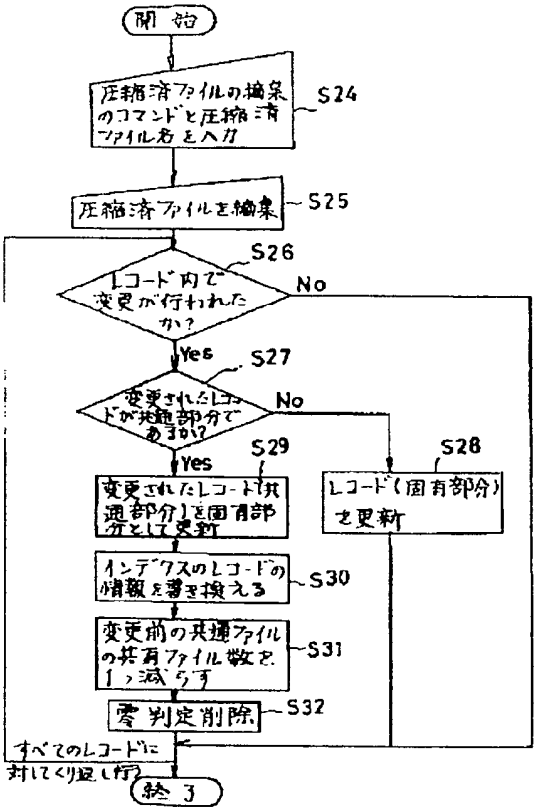
【図8】


開始
削除のコマンドと削除する圧縮済みファイルの名前を入力
S12
インデクスにアクセスしてレコードの情報を読み取る
S13
S14
レコードが共通部分であるか?
No
Yes
S15
固有部分を削除
共通ファイルの共通部分制御語の共有ファイル数を1つ減らす
S16
零判定削除
S17
すべてのレコードに対して繰り返し行う
S18
インデクスを削除
終了

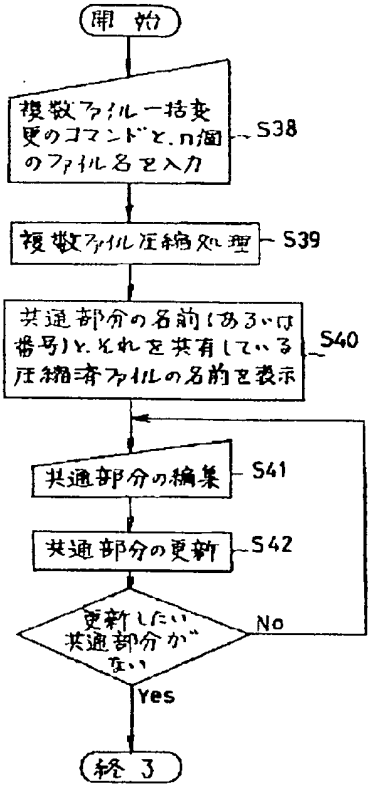
【図11】


開始
複数ファイル圧縮処理のコマンドとn個のファイル名を入力する
S33
i←n
S34
i番のファイル間で既に共通部分であると判定された部分以外との内容を比較
S35
共通部分があったか?
No
Yes
共通部分iに対する共通部分処理
S36
nC1回くり返す
i←i-1
No
i=m
Yes
比較後ファイル処理
S37
n回くり返す
終了

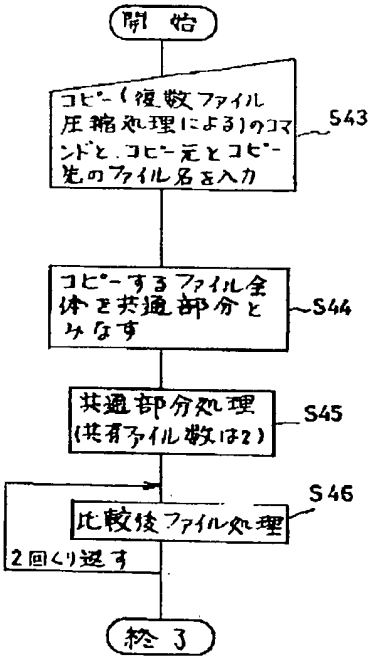
【図10】


開始
圧縮済ファイルの編集のコマンドと圧縮済ファイル名を入力
S24
圧縮済ファイルを編集
S25
S26
レコード内で変更が行われたか?
No
Yes
S27
変更されたレコードが共通部分であるか?
No
Yes
S29
S28
変更されたレコード[共通部分]を固有部分として更新
レコード(固有部分)を更新
インデックスのレコードの情報を書き換える
S30
変更前の共通ファイルの共有ファイル数を1つ減らす
S31
零判定削除
S32
すべてのレコードに対してくり返し行う
終了

【図12】


開始
複数ファイル一括変更のコマンドとn個のファイル名を入力
S38
複数ファイル圧縮処理
S39
共通部分の名前(あるいは番号)とそれを共有している圧縮済ファイルの名前を表示
S40
共通部分の編集
S41
共通部分の更新
S42
更新したい共通部分がない
No
Yes
終了

【図13】


開始
コピー(複数ファイル圧縮処理による)のコマンドとコピー元とコピー先のファイル名を入力
S43
コピーするファイル全体を共通部分とみなす
S44
共通部分処理(共有ファイル数は2)
S45
比較後ファイル処理
S46
2回くり返す
終了